

2016-03-09

Making new discoveries from old data: Utilizing digital scholarship to foster new research in Special Collections

Carruthers, Matthew

<https://hdl.handle.net/2027.42/138108>

<http://creativecommons.org/licenses/by-nc-sa/4.0/>

Downloaded from Deep Blue, University of Michigan's institutional repository

Making new discoveries from old data

Utilizing digital scholarship to foster new research in
special collections

Slides: goo.gl/smvkvf

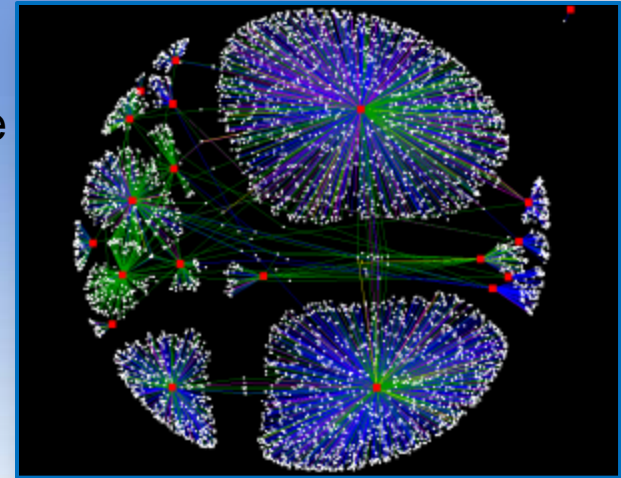
Goals

1. Explore using digital scholarship methods and tools to enhance user experience in special collections and potentially provide new services.
2. Develop a pilot project as a proof of concept for a new service.
 - Utilize existing open source technologies.
 - Little to no need for dedicated IT support.

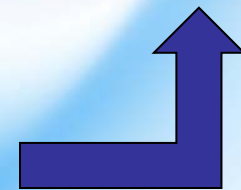
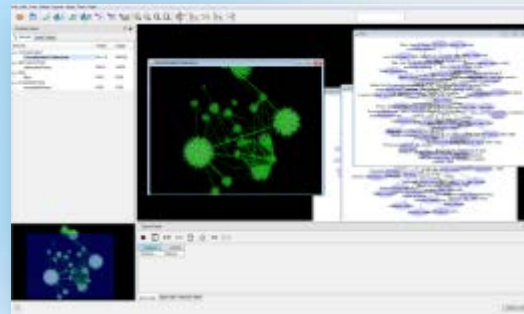
Result

Provide visualization on demand, customized to a patron's individual research question.

- Use data from finding aids and publicly accessible databases along with open-source tools to create interactive visualizations of social networks of the creators of our archival collections.
- Make connections between entities more apparent and provide a discovery tool to spur further inquiry and research.



```
1 # Create a graph from the data
2 graph = nx.Graph()
3
4 # Add nodes
5 for i in range(0, len(data)):
6     node = data[i]
7     graph.add_node(node)
8
9 # Add edges
10 for i in range(0, len(data)):
11     node = data[i]
12     for j in range(i+1, len(data)):
13         node2 = data[j]
14         if node < node2:
15             graph.add_edge(node, node2)
```



Data Processing

EAD

[illegible]

EAC-CPF

[illegible]

Encoded Archival Description (EAD)

- Metadata standard for the encoding of finding aids for use in a networked (online) environment.

Encoded Archival Context – Corporate Bodies, Persons, and Families (EAC-CPF)

- Companion standard to Encoded Archival Description (EAD).
- Describes the entity responsible for creating (or who is represented in) one or more archival collections.

Data Processing: Tools

EAC-CPF record creation:

Social Networks and Archival Context

<http://socialarchive.iath.virginia.edu/snac/search>

Remixing Archival Metadata Project

<http://rampeditor.info>

xEAC

<https://github.com/ewg118/xEAC>

More Data Processing

Use XSLT to extract information about connections and format output files as tab-delimited text.

EAC-CPF

XSLT

Tab-delimited text

[illegible]

```

100 # Create a new instance of the class
101 my_instance = MyClass()
102
103 # Call the method
104 my_instance.my_method()
105
106 # Print the output
107 print(my_instance.my_method())
108
109 # Create a new instance of the class
110 my_instance2 = MyClass()
111
112 # Call the method
113 my_instance2.my_method()
114
115 # Print the output
116 print(my_instance2.my_method())
117
118 # Create a new instance of the class
119 my_instance3 = MyClass()
120
121 # Call the method
122 my_instance3.my_method()
123
124 # Print the output
125 print(my_instance3.my_method())
126
127 # Create a new instance of the class
128 my_instance4 = MyClass()
129
130 # Call the method
131 my_instance4.my_method()
132
133 # Print the output
134 print(my_instance4.my_method())
135
136 # Create a new instance of the class
137 my_instance5 = MyClass()
138
139 # Call the method
140 my_instance5.my_method()
141
142 # Print the output
143 print(my_instance5.my_method())
144
145 # Create a new instance of the class
146 my_instance6 = MyClass()
147
148 # Call the method
149 my_instance6.my_method()
150
151 # Print the output
152 print(my_instance6.my_method())
153
154 # Create a new instance of the class
155 my_instance7 = MyClass()
156
157 # Call the method
158 my_instance7.my_method()
159
160 # Print the output
161 print(my_instance7.my_method())
162
163 # Create a new instance of the class
164 my_instance8 = MyClass()
165
166 # Call the method
167 my_instance8.my_method()
168
169 # Print the output
170 print(my_instance8.my_method())
171
172 # Create a new instance of the class
173 my_instance9 = MyClass()
174
175 # Call the method
176 my_instance9.my_method()
177
178 # Print the output
179 print(my_instance9.my_method())
180
181 # Create a new instance of the class
182 my_instance10 = MyClass()
183
184 # Call the method
185 my_instance10.my_method()
186
187 # Print the output
188 print(my_instance10.my_method())
189
190 # Create a new instance of the class
191 my_instance11 = MyClass()
192
193 # Call the method
194 my_instance11.my_method()
195
196 # Print the output
197 print(my_instance11.my_method())
198
199 # Create a new instance of the class
200 my_instance12 = MyClass()
201
202 # Call the method
203 my_instance12.my_method()
204
205 # Print the output
206 print(my_instance12.my_method())
207
208 # Create a new instance of the class
209 my_instance13 = MyClass()
210
211 # Call the method
212 my_instance13.my_method()
213
214 # Print the output
215 print(my_instance13.my_method())
216
217 # Create a new instance of the class
218 my_instance14 = MyClass()
219
220 # Call the method
221 my_instance14.my_method()
222
223 # Print the output
224 print(my_instance14.my_method())
225
226 # Create a new instance of the class
227 my_instance15 = MyClass()
228
229 # Call the method
230 my_instance15.my_method()
231
232 # Print the output
233 print(my_instance15.my_method())
234
235 # Create a new instance of the class
236 my_instance16 = MyClass()
237
238 # Call the method
239 my_instance16.my_method()
240
241 # Print the output
242 print(my_instance16.my_method())
243
244 # Create a new instance of the class
245 my_instance17 = MyClass()
246
247 # Call the method
248 my_instance17.my_method()
249
250 # Print the output
251 print(my_instance17.my_method())
252
253 # Create a new instance of the class
254 my_instance18 = MyClass()
255
256 # Call the method
257 my_instance18.my_method()
258
259 # Print the output
260 print(my_instance18.my_method())
261
262 # Create a new instance of the class
263 my_instance19 = MyClass()
264
265 # Call the method
266 my_instance19.my_method()
267
268 # Print the output
269 print(my_instance19.my_method())
270
271 # Create a new instance of the class
272 my_instance20 = MyClass()
273
274 # Call the method
275 my_instance20.my_method()
276
277 # Print the output
278 print(my_instance20.my_method())
279
280 # Create a new instance of the class
281 my_instance21 = MyClass()
282
283 # Call the method
284 my_instance21.my_method()
285
286 # Print the output
287 print(my_instance21.my_method())
288
289 # Create a new instance of the class
290 my_instance22 = MyClass()
291
292 # Call the method
293 my_instance22.my_method()
294
295 # Print the output
296 print(my_instance22.my_method())
297
298 # Create a new instance of the class
299 my_instance23 = MyClass()
300
301 # Call the method
302 my_instance23.my_method()
303
304 # Print the output
305 print(my_instance23.my_method())
306
307 # Create a new instance of the class
308 my_instance24 = MyClass()
309
310 # Call the method
311 my_instance24.my_method()
312
313 # Print the output
314 print(my_instance24.my_method())
315
316 # Create a new instance of the class
317 my_instance25 = MyClass()
318
319 # Call the method
320 my_instance25.my_method()
321
322 # Print the output
323 print(my_instance25.my_method())
324
325 # Create a new instance of the class
326 my_instance26 = MyClass()
327
328 # Call the method
329 my_instance26.my_method()
330
331 # Print the output
332 print(my_instance26.my_method())
333
334 # Create a new instance of the class
335 my_instance27 = MyClass()
336
337 # Call the method
338 my_instance27.my_method()
339
340 # Print the output
341 print(my_instance27.my_method())
342
343 # Create a new instance of the class
344 my_instance28 = MyClass()
345
346 # Call the method
347 my_instance28.my_method()
348
349 # Print the output
350 print(my_instance28.my_method())
351
352 # Create a new instance of the class
353 my_instance29 = MyClass()
354
355 # Call the method
356 my_instance29.my_method()
357
358 # Print the output
359 print(my_instance29.my_method())
360
361 # Create a new instance of the class
362 my_instance30 = MyClass()
363
364 # Call the method
365 my_instance30.my_method()
366
367 # Print the output
368 print(my_instance30.my_method())
369
370 # Create a new instance of the class
371 my_instance31 = MyClass()
372
373 # Call the method
374 my_instance31.my_method()
375
376 # Print the output
377 print(my_instance31.my_method())
378
379 # Create a new instance of the class
380 my_instance32 = MyClass()
381
382 # Call the method
383 my_instance32.my_method()
384
385 # Print the output
386 print(my_instance32.my_method())
387
388 # Create a new instance of the class
389 my_instance33 = MyClass()
390
391 # Call the method
392 my_instance33.my_method()
393
394 # Print the output
395 print(my_instance33.my_method())
396
397 # Create a new instance of the class
398 my_instance34 = MyClass()
399
400 # Call the method
401 my_instance34.my_method()
402
403 # Print the output
404 print(my_instance34.my_method())
405
406 # Create a new instance of the class
407 my_instance35 = MyClass()
408
409 # Call the method
410 my_instance35.my_method()
411
412 # Print the output
413 print(my_instance35.my_method())
414
415 # Create a new instance of the class
416 my_instance36 = MyClass()
417
418 # Call the method
419 my_instance36.my_method()
420
421 # Print the output
422 print(my_instance36.my_method())
423
424 # Create a new instance of the class
425 my_instance37 = MyClass()
426
427 # Call the method
428 my_instance37.my_method()
429
430 # Print the output
431 print(my_instance37.my_method())
432
433 # Create a new instance of the class
434 my_instance38 = MyClass()
435
436 # Call the method
437 my_instance38.my_method()
438
439 # Print the output
440 print(my_instance38.my_method())
441
442 # Create a new instance of the class
443 my_instance39 = MyClass()
444
445 # Call the method
446 my_instance39.my_method()
447
448 # Print the output
449 print(my_instance39.my_method())
450
451 # Create a new instance of the class
452 my_instance40 = MyClass()
453
454 # Call the method
455 my_instance40.my_method()
456
457 # Print the output
458 print(my_instance40.my_method())
459
460 # Create a new instance of the class
461 my_instance41 = MyClass()
462
463 # Call the method
464 my_instance41.my_method()
465
466 # Print the output
467 print(my_instance41.my_method())
468
469 # Create a new instance of the class
470 my_instance42 = MyClass()
471
472 # Call the method
473 my_instance42.my_method()
474
475 # Print the output
476 print(my_instance42.my_method())
477
478 # Create a new instance of the class
479 my_instance43 = MyClass()
480
481 # Call the method
482 my_instance43.my_method()
483
484 # Print the output
485 print(my_instance43.my_method())
486
487 # Create a new instance of the class
488 my_instance44 = MyClass()
489
490 # Call the method
491 my_instance44.my_method()
492
493 # Print the output
494 print(my_instance44.my_method())
495
496 # Create a new instance of the class
497 my_instance45 = MyClass()
498
499 # Call the method
500 my_instance45.my_method()
501
502 # Print the output
503 print(my_instance45.my_method())
504
505 # Create a new instance of the class
506 my_instance46 = MyClass()
507
508 # Call the method
509 my_instance46.my_method()
510
511 # Print the output
512 print(my_instance46.my_method())
513
514 # Create a new instance of the class
515 my_instance47 = MyClass()
516
517 # Call the method
518 my_instance47.my_method()
519
520 # Print the output
521 print(my_instance47.my_method())
522
523 # Create a new instance of the class
524 my_instance48 = MyClass()
525
526 # Call the method
527 my_instance48.my_method()
528
529 # Print the output
530 print(my_instance48.my_method())
531
532 # Create a new instance of the class
533 my_instance49 = MyClass()
534
535 # Call the method
536 my_instance49.my_method()
537
538 # Print the output
539 print(my_instance49.my_method())
540
541 # Create a new instance of the class
542 my_instance50 = MyClass()
543
544 # Call the method
545 my_instance50.my_method()
546
547 # Print the output
548 print(my_instance50.my_method())
549
550 # Create a new instance of the class
551 my_instance51 = MyClass()
552
553 # Call the method
554 my_instance51.my_method()
555
556 # Print the output
557 print(my_instance51.my_method())
558
559 # Create a new instance of the class
560 my_instance52 = MyClass()
561
562 # Call the method
563 my_instance52.my_method()
564
565 # Print the output
566 print(my_instance52.my_method())
567
568 # Create a new instance of the class
569 my_instance53 = MyClass()
570
571 # Call the method
572 my_instance53.my_method()
573
574 # Print the output
575 print(my_instance53.my_method())
576
577 # Create a new instance of the class
578 my_instance54 = MyClass()
579
580 # Call the method
581 my_instance54.my_method()
582
583 # Print the output
584 print(my_instance54.my_method())
585
586 # Create a new instance of the class
587 my_instance55 = MyClass()
588
589 # Call the method
590 my_instance55.my_method()
591
592 # Print the output
593 print(my_instance55.my_method())
594
595 # Create a new instance of the class
596 my_instance56 = MyClass()
597
598 # Call the method
599 my_instance56.my_method()
600
601
```

[illegible]

More Data Processing

Information extracted:

- Name of connection
- Type of entity
- Type of connection
- Link to more information (when available)

```
<cpfRelation
  xlink:arcrole="http://socialarchive.iath.virginia.edu/control/term#associatedWith"
  xlink:href="http://n2t.net/ark:/99166/w6f48mrk"
  xlink:role="http://socialarchive.iath.virginia.edu/control/term#Person"
  xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink">
  <relationEntry>Zirlin, Larry.</relationEntry>
</cpfRelation>
```


More Data Processing

Information extracted:

- Name of connection
- Type of entity
- Type of connection
- Link to more information (when available)

```
<cpfRelation
  xlink:arcrole="http://socialarchive.iath.virginia.edu/control/term#associatedWith"
  xlink:href="http://n2t.net/ark:/99166/w6f48mrk"
  xlink:role="http://socialarchive.iath.virginia.edu/control/term#Person"
  xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink">
  <relationEntry>Zirlin, Larry.</relationEntry>
</cpfRelation>
```

More Data Processing

Information extracted:

- Name of connection
- **Type of entity**
- Type of connection
- Link to more information (when available)

Entity types:

- Person
- Corporate Body
- Family
- Archival Resource
- Bibliographic Resource

```
<cpfRelation
  xlink:arcrole="http://socialarchive.iath.virginia.edu/control/term#associatedWith"
  xlink:href="http://n2t.net/ark:/99166/w6f48mrk"
  xlink:role="http://socialarchive.iath.virginia.edu/control/term#Person"
  xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink">
  <relationEntry>Zirlin, Larry.</relationEntry>
</cpfRelation>
```

The Process

Information extracted:

- Name of connection
- Type of entity
- **Type of connection**
- Link to more information (when available)

Connection types:

- Associated With
- Corresponded With
- Creator Of
- Referenced In

```
<cpfRelation
  xlink:arcrole="http://socialarchive.iath.virginia.edu/control/term#associatedWith"
  xlink:href="http://n2t.net/ark:/99166/w6f48mrk"
  xlink:role="http://socialarchive.iath.virginia.edu/control/term#Person"
  xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink">
  <relationEntry>Zirlin, Larry.</relationEntry>
</cpfRelation>
```

More Data Processing

Information extracted:

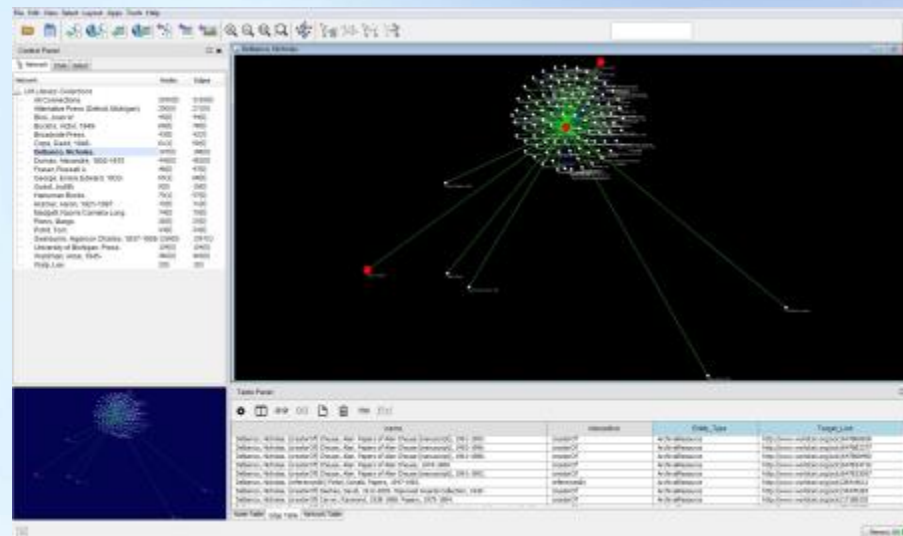
- Name of connection
- Type of entity
- Type of connection
- Link to more information (when available)

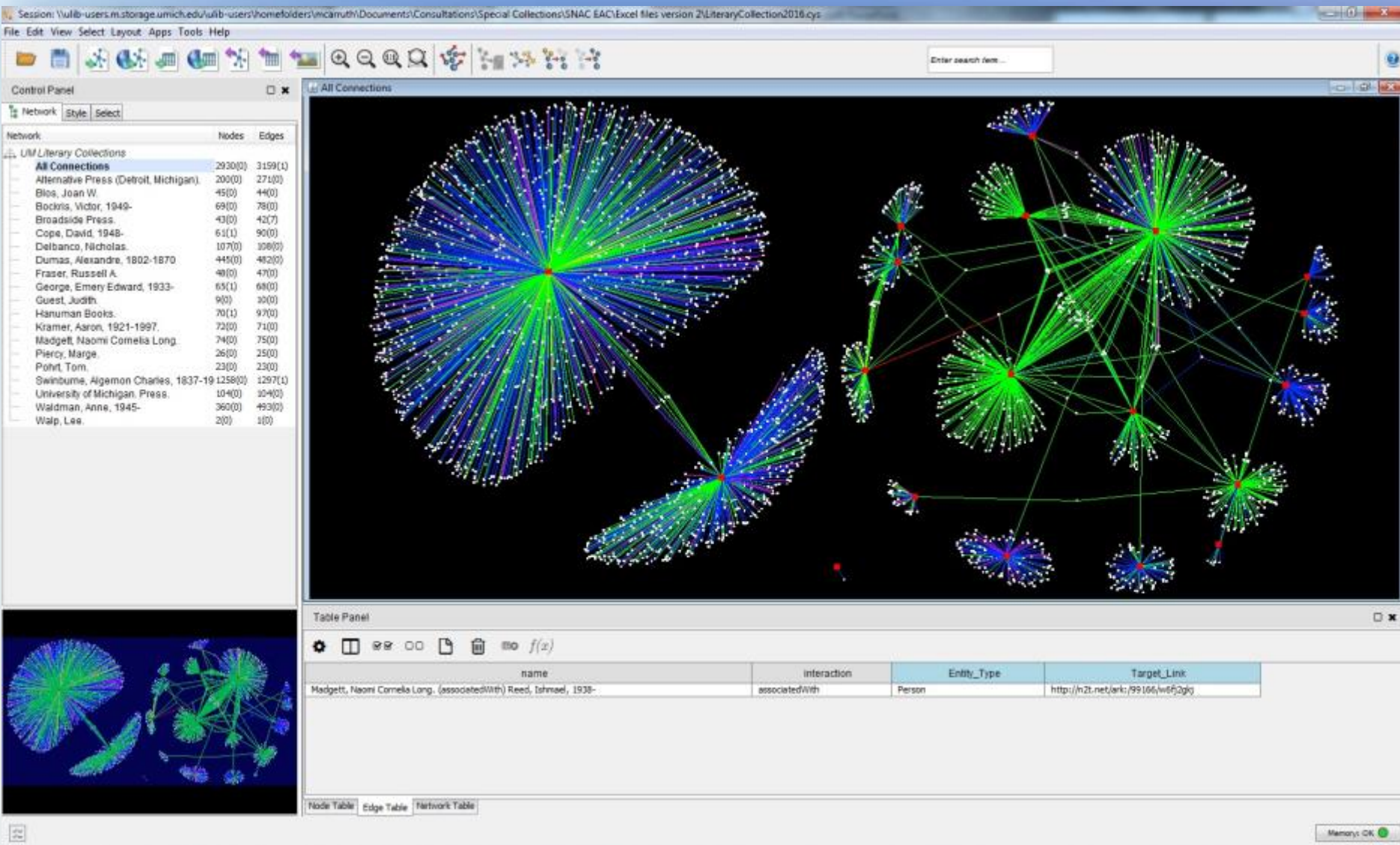
```
<cpfRelation
  xlink:arcrole="http://socialarchive.iath.virginia.edu/control/term#associatedWith"
  xlink:href="http://n2t.net/ark:/99166/w6f48mrk"
  xlink:role="http://socialarchive.iath.virginia.edu/control/term#Person"
  xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink">
  <relationEntry>Zirlin, Larry.</relationEntry>
</cpfRelation>
```

Visualizing the Data

Cytoscape – <http://cytoscape.org/>





- Open source desktop application for visualizing complex networks.

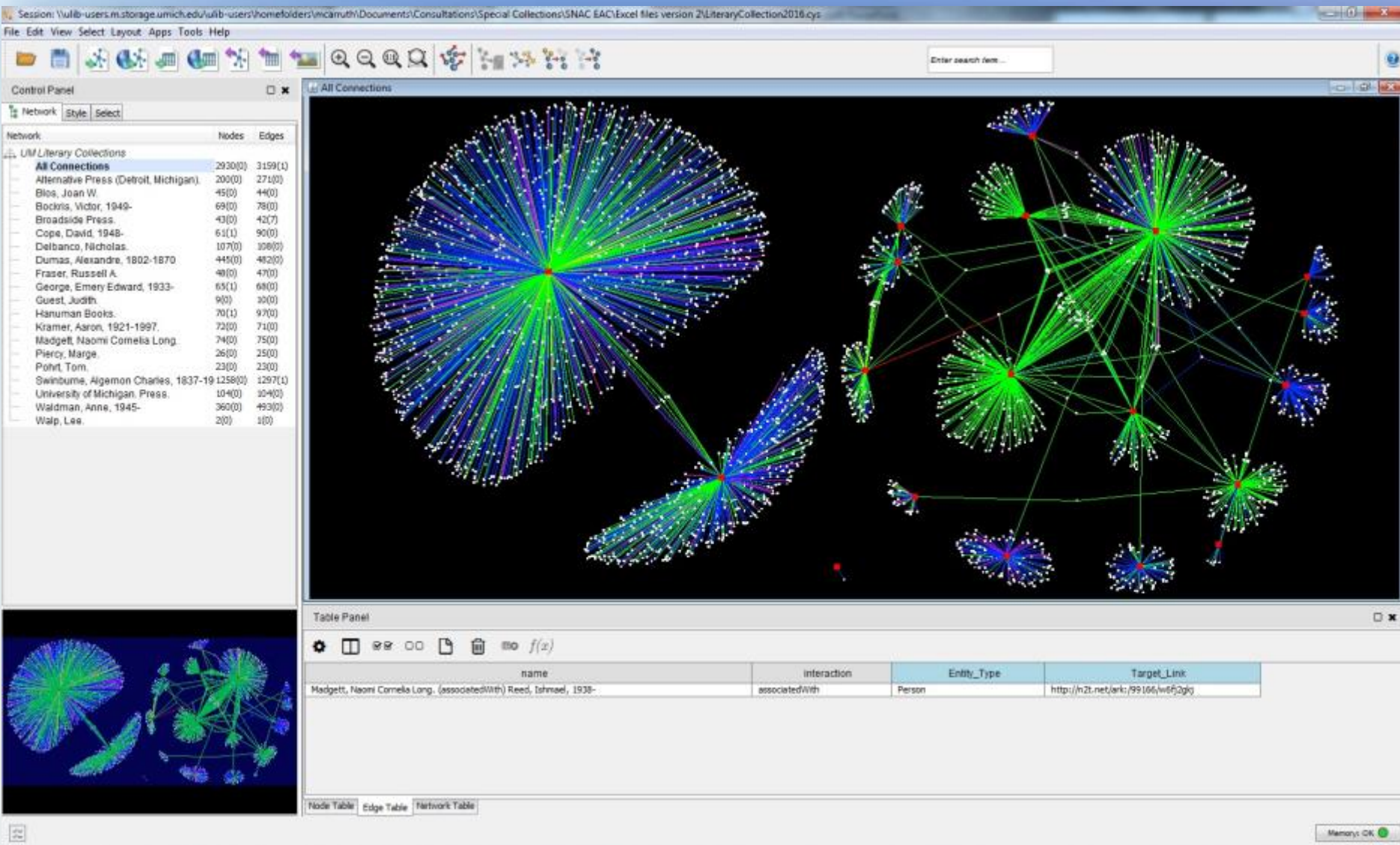




Visualizing the Data

Edge Color (Unselected) Mapping

Edge Color (Unselected)	interaction
	associatedWith
	correspondedWith
	creatorOf
	referencedIn



Code4Lib 2016
March 9, 2016



Enter search term...

Control Panel

Network	Nodes	Edges
UM Library Collections		
All Connections	2930(0)	3159(1)
Alternative Press (Detroit, Michigan)	200(0)	271(0)
Blos, Joan W.	45(0)	44(0)
Bockris, Victor, 1949-	69(0)	78(0)
Broadside Press	43(0)	42(7)
Cope, David, 1948-	61(1)	90(0)
Delbanco, Nicholas	207(0)	108(0)
Dumas, Alexandre, 1802-1870	445(0)	482(0)
Fraser, Russell A.	48(0)	47(0)
George, Emery Edward, 1933-	65(1)	68(0)
Guest, Judith	9(0)	10(0)
Hanuman Books	70(1)	97(0)
Kramer, Aaron, 1921-1997	72(0)	71(0)
Madgett, Naomi Cornelia Long	74(0)	75(0)
Piercy, Marge	26(0)	25(0)
Pohrt, Tom	23(0)	23(0)
Swinburne, Algernon Charles, 1837-19	1258(0)	1297(1)
University of Michigan Press	304(0)	104(0)
Waldman, Anne, 1945-	360(0)	493(0)
Walp, Lee	2(0)	1(0)

Delbanco, Nicholas

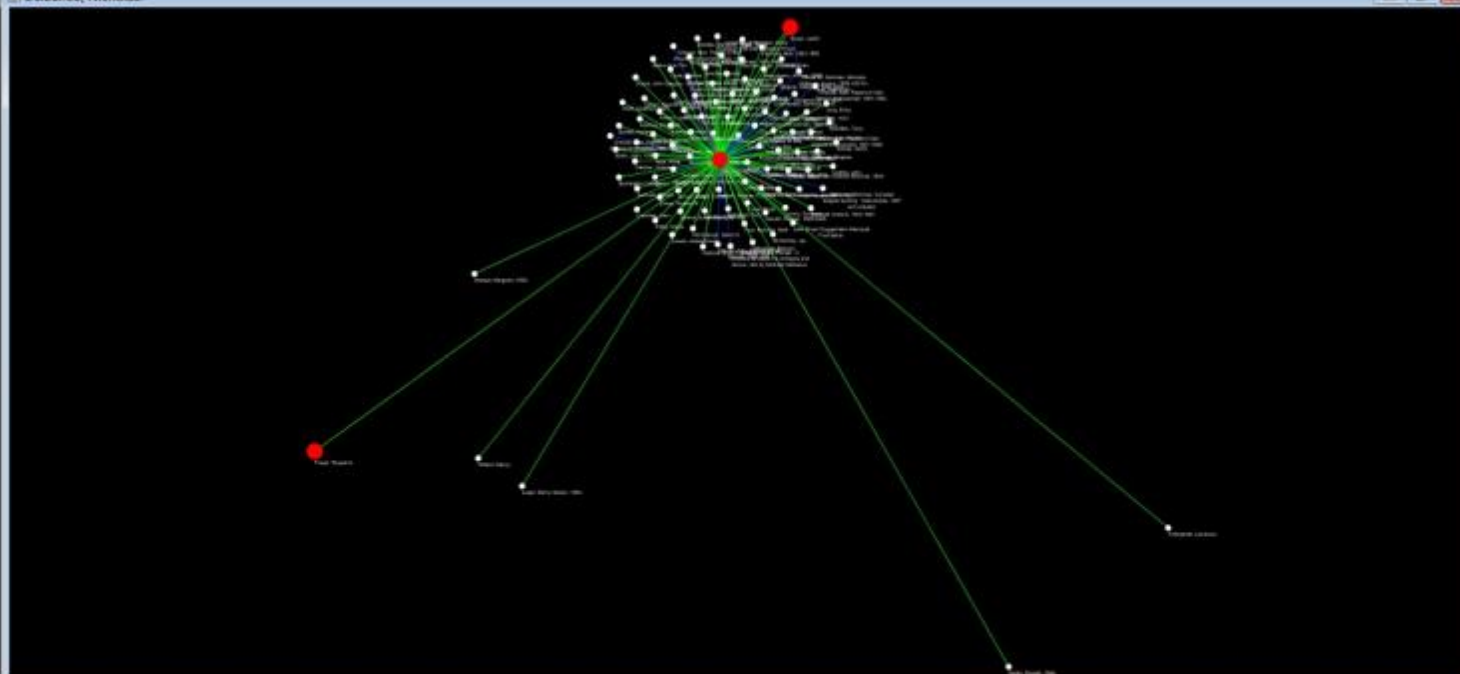
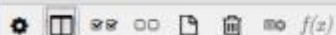


Table Panel



name	Interaction	Entity_Type	Target_Link
Delbanco, Nicholas. (creatorOf) Cheuse, Alan. Papers of Alan Cheuse [manuscript]. 1991-1995.	creatorOf	ArchivalResource	http://www.worldcat.org/oclc/647860658
Delbanco, Nicholas. (creatorOf) Cheuse, Alan. Papers of Alan Cheuse [manuscript]. 1992-1996.	creatorOf	ArchivalResource	http://www.worldcat.org/oclc/647883377
Delbanco, Nicholas. (creatorOf) Cheuse, Alan. Papers of Alan Cheuse [manuscript]. 1961-1988.	creatorOf	ArchivalResource	http://www.worldcat.org/oclc/647808490
Delbanco, Nicholas. (creatorOf) Cheuse, Alan. Papers of Alan Cheuse [manuscript]. 1974-1989.	creatorOf	ArchivalResource	http://www.worldcat.org/oclc/647824710
Delbanco, Nicholas. (creatorOf) Cheuse, Alan. Papers of Alan Cheuse [manuscript]. 1991-1992.	creatorOf	ArchivalResource	http://www.worldcat.org/oclc/647833097
Delbanco, Nicholas. (referencedIn) Finkel, Donald. Papers, 1947-1985.	referencedIn	ArchivalResource	http://www.worldcat.org/oclc/28419612
Delbanco, Nicholas. (creatorOf) Gaches, David, 1912-2005. Hopwood Awards Collection, 1930-	creatorOf	ArchivalResource	http://www.worldcat.org/oclc/34370265
Delbanco, Nicholas. (creatorOf) Carver, Raymond, 1938-1988. Papers, 1975-1984.	creatorOf	ArchivalResource	http://www.worldcat.org/oclc/17188335

Node Table Edge Table Network Table

Memory OK

Sharing the Visualizations

Share visualizations via web-publishing companion platform to Cytoscape:

CyNetShare

<http://cynetshare.ucsd.edu/#/>

Working example:

<http://goo.gl/eHYLUi>

Sharing the Visualizations

CyNetShare



An easy way to share your network visualizations

Project Web Site

Documentation

Public Files

Gists

Network Data URL

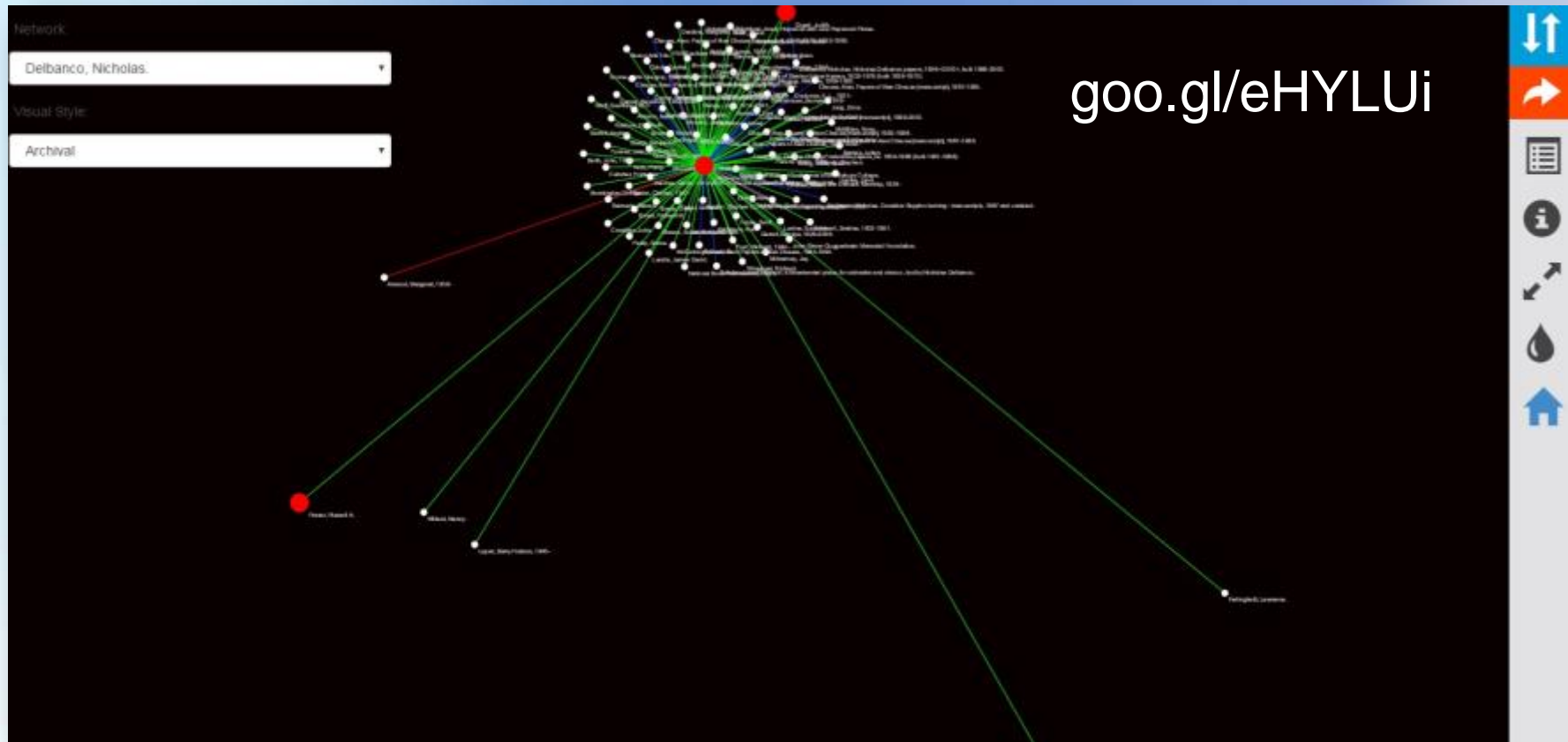
Enter Network File URL

Visual Style URL
(Optional)

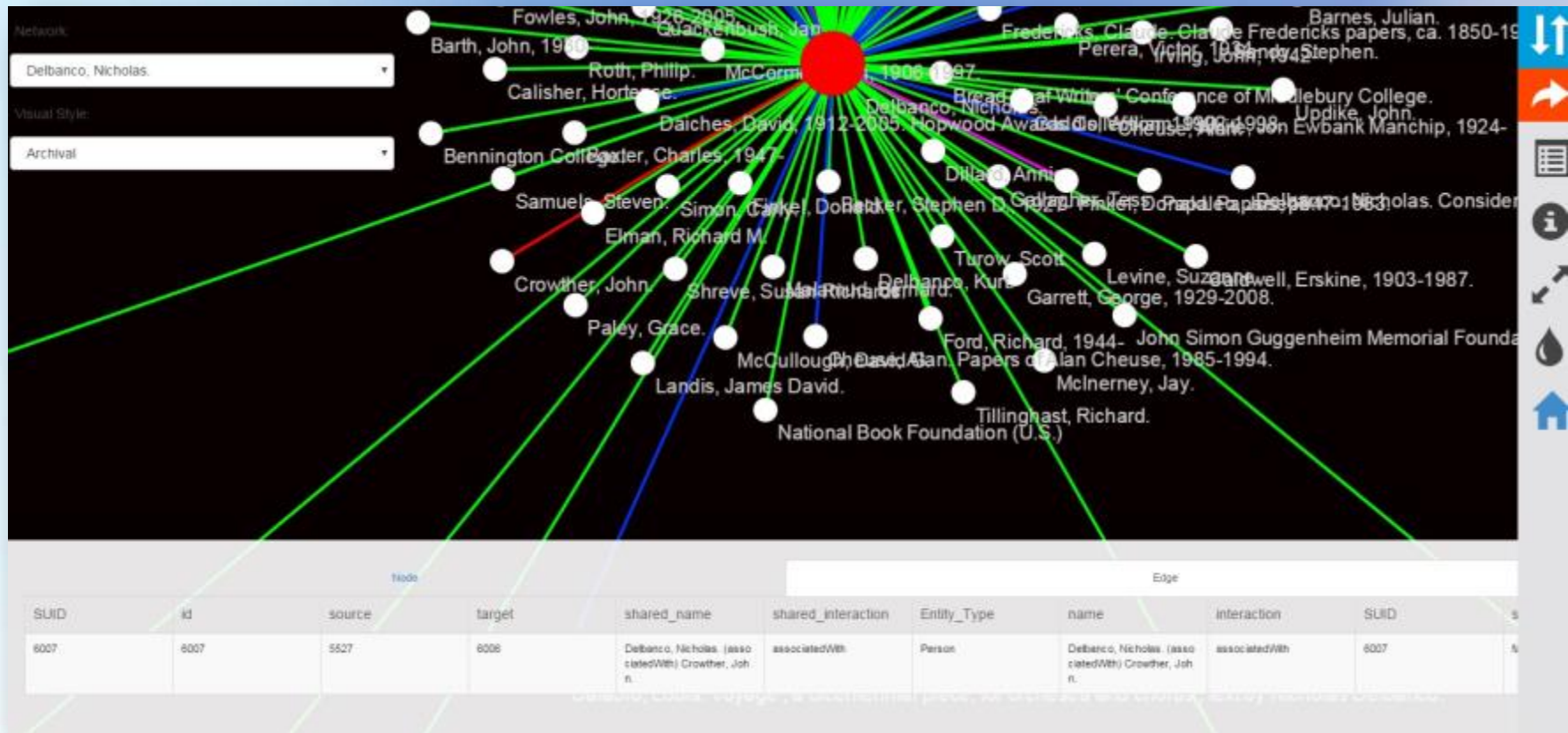
Enter Style URL

Visualize!

Sharing the Visualizations



Sharing the Visualizations



Next Steps

Continue to streamline the workflow

Next Steps

Perform user testing

(Incorporate feedback into service model)

Next Steps

Explore more options for using and
manipulating the data

(Customize and host local CyNetShare instance)

[http://www-
personal.umich.edu/~mcarruth/HanumanBooksVis/web_session/index.html#/](http://www-personal.umich.edu/~mcarruth/HanumanBooksVis/web_session/index.html#/)

Thanks!

Matt Carruthers

mcarruth@umich.edu

@mattadata2

Slides: goo.gl/smvkvf